

(23.06.04)

IB04/2141



Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

REC'D 06 JUL 2004

WIPO

PCT

Bescheinigung

Certificate

Attestation

Die angehefteten Unterla-
gen stimmen mit der
ursprünglich eingereichten
Fassung der auf dem näch-
sten Blatt bezeichneten
europäischen Patentanmel-
dung überein.

The attached documents
are exact copies of the
European patent application
described on the following
page, as originally filed.

Les documents fixés à
cette attestation sont
conformes à la version
initialement déposée de
la demande de brevet
européen spécifiée à la
page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

03300050.6

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

BEST AVAILABLE COPY



Anmeldung Nr:
Application no.: 03300050.6
Demande no:

Anmeldetag:
Date of filing: 04.07.03
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

Koninklijke Philips Electronics N.V.
Groenewoudseweg 1
5621 BA Eindhoven
PAYS-BAS

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

METHOD FOR DOWNLOADING A BROADCASTED MULTIMEDIA CONTENT OVER A DISTRIBUTION
NETWORK

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s)
revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

H04N7/00

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of
filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL
PT RO SE SI SK TR LI

DESCRIPTION

Field of the invention

The present invention relates to a method for downloading a multimedia content, for instance a live event, to a client device. The invention also relates to a network, a system, a slicer, a server, and a client device specifically designed to implement such a
5 downloading method.

The invention has interesting applications for the transmission of live multimedia contents (like live events or broadcasted TV programs) to client devices via the Internet. It also applies to the transmission of video from one mobile device to one or more peer devices via the Internet.
10

Background of the invention

The document "An RTP to HTTP Video Gateway" by Mathias Johanson, WWW10, May 1-5 2001, Hong-Kong (ACM, 1-58113-348-0/01/0005) relates to the transmission of videoconferencing content through the Internet.

Typically when live multimedia contents are transmitted through an IP network, the transport protocol that is used is RTP (Real time Transport Protocol) over UDP (User Datagram Protocol). RTP is defined in the RFC1889 published by the IETF (Internet Engineering Task Force). UDP is a connection less transport protocol defined in the RFC0768 of the IETF.
15

As stated in the above-mentioned document, a main problem when using RTP/UDP is that generally firewalls don't allow UDP traffic to pass through and that in many cases they employ techniques like Network Address Translation (sometimes referred to as NAT) that complicate end-to-end real time communication. As a consequence, the vast majority of Internet users are not able to receive live multimedia contents like videoconferencing content.
20
25

The solution proposed in this document is specifically designed for videoconferencing applications. It consists in encoding the video content into a sequence of JPEG images and in describing the sequence of JPEG images as a MIME multipart/mixed message where each part of the message is of the image/JPEG type (MIME stands for Multipurpose Internet Mail Extension; it describes a format for electronic messages). When the client browser is recognized as not capable of displaying
30

MIME multipart/mixed content, a specific java applet is downloaded to the client device to ensure correct display of the sequence of JPEG images.

The solution proposed in this document may be satisfactory for certain types of applications, in particular for web-camera applications. However it is not adapted to the transmission of large multimedia contents like live events or movies for the reasons stated below:

- In the proposed solution, the video is encoded as a sequence of JPEG images. Each image is encoded independently so that the temporal redundancy between images is not exploited. Therefore the compression ratio is low. It is definitively too low to allow for the transmission of large video contents via low bit rate connections like Internet connections and/or mobile networks.

- The proposed solution doesn't provide an appropriate solution for handling the audio part of the video content. In particular synchronizing the images with the audio is not straightforward.

Another drawback of the proposed solution is that it is based on the MIME multipart message format which, in practice, is not supported by most of the client browsers.

One of the objects of the invention is to propose an efficient solution for transmitting live multimedia contents in real time from a server to a client device.

Summary of the invention

This is achieved with a network as claimed in claim 1, a system as claimed in claim 2, a slicer as claimed in claim 3, a server as claimed in claims 4 to 7, a client device as claimed in claims 8 and 9, and a downloading method as claimed in claim 10.

A network according to the invention comprises:

- a source for acquiring a multimedia content,
- an encoder for encoding said multimedia content,
- a slicer for slicing said encoded multimedia content in at least one set of slicing positions so as to generate at least one set of slices where each slice can be decoded independently from the others, and for providing at least one set of files, each file containing a slice of said encoded multimedia content,
- a distribution network,
- an access provider for providing a client device with an access to said distribution network,

- a server linked to said distribution network and having access to said set(s) of files for downloading at least one of said files to said client device via said distribution network upon reception of an initial request from said client device, said request being directed to said encoded multimedia content.

5 The invention proposes to transform a multimedia content (for instance a broadcasted TV program, a broadcasted live event, or a personal video capture with a mobile device) into a file-based content. More specifically, the proposed solution consists in slicing an encoded content into a plurality of slices where each slice comprises a given amount of time of the content and can be decoded independently from
10 the other slices. Each slice is stored in a file that can be downloaded as soon as it is ready at the server side and can be played as soon as it is received at the client side.

 According to the invention the file-based content is transmitted from the server to the client device via a point-to-point connection. On IP networks, point-to-point connections are usually ruled by the HTTP protocol (Hyper Text Transfer Protocol,
15 defined in the RFC2616 of the IETF). The HTTP protocol is the basis for the World Wide Web and therefore has the great advantage of being accepted by all firewalls and Networks Address Translators. This means that the transmitted content will be accessible for any client device having access to the World Wide Web without restriction.

20 The invention proposes to generate one or more set of files for the same multimedia content. When several sets of files are generated, the slicing positions vary from one set to another (the files are starting at different positions in time in each set). When a client sends an initial request, he receives either the previous file -which means he will receive out of date information- or he has to wait for the next file to be ready. In
25 both cases, using several sets of files allows reducing the inconvenience for the user. The number of sets is a compromise between the inconvenience experienced by the client and the storage unit space needed to store the files.

 According to the invention it is possible to download one file only, upon reception of an initial request by a client device. This is advantageous for certain
30 applications, for instance to allow clients to get a quick outlook of the results during championships.

 In a first embodiment, the server has:

- means for sending a document to said client device upon reception of said initial request, said document causing said client device to repetitively send a fetching request

designating said encoded multimedia content,

- means for selecting which file is to be downloaded amongst said set(s) of files, upon reception of said fetching requests from said client device,

- means for downloading the selected file.

5 The files can be either fetched one by one by the client device or sent one by one by the server upon reception of an initial request. In practice, it is not certain that all client browsers will support receiving several files in response to one single request. Therefore it will usually be preferred that the client device fetches the files one by one (i.e. sends a fetching request for each file to be downloaded).

10 The client device can be designed specifically so as to send automatically the fetching request at the appropriate time. But it is usually preferred to use standard client devices. Therefore it is advantageous to send a document from the server to the client device, said document causing the client device to repetitively send a fetching request.

Advantageously, said document comprises an instruction for the client device to
15 send a subsequent fetching request a certain amount of time before the end of the playback of the previous file. In such a way, it is made sure that the next file will reach the client device early enough, and that the client will not experience any gap in the playback process.

In a second alternative embodiment, the server has:

20 - means for sending a document to said client device upon reception of said initial request directed to an encoded multimedia content, said document comprising a resource identifier designating said encoded multimedia content and specific to said client device, said document causing said client device to repetitively send a fetching request containing said resource identifier,

25 - means, activated upon reception of a first fetching request, for:
selecting a first file amongst said set(s) of files, downloading the selected file, and keeping a record of said resource identifier together with an indication of which file was downloaded,

- means, activated upon reception of subsequent fetching requests, for:
30 checking said record so as to select the file that follows the one that was previously downloaded, downloading the selected file, and updating said record.

HTTP is a stateless protocol and therefore HTTP requests issued by the same client device are normally processed independently one from the other. This second embodiment is advantageously used when several sets of files are generated for the same

multimedia content so as to allow the server to keep track of each client. As the server knows which file is to be sent next for each client, smooth playback of the content can be achieved.

5 Brief description of the figures

These and other aspects of the invention are further described by reference to the following drawings:

- figure 1 is a schematic representation of a first example of network according to the invention,
- 10 - figure 2 is a schematic representation of a plurality of sets of files generated by a slicer according to the invention,
- figure 3 is a schematic representation of two overlapping files,
- figure 4 is schematic representation of three sets of files,
- figure 5 is a block diagram of a method according to the invention for downloading a
- 15 live multimedia content,
- figures 6 and 7 are schematic representations of other examples of a network according to the invention.

Description of preferred embodiment

- 20 Figure 1 is a schematic representation of a network according to the invention. The network of figure 1 comprises:
- a source 1 for acquiring a multimedia content;
 - a broadcaster 2 for broadcasting said multimedia content;
 - a receiver 3 for receiving a broadcasted multimedia content;
 - 25 - a system 4 comprising an encoder 5 for encoding a received multimedia content, and a slicer 6 for slicing an encoded multimedia content in at least one set of slicing positions so as generate at least one set of slices where each slice can be decoded independently from the others, and for providing at least one set of files, each file containing a slice of said encoded multimedia content,
 - 30 - a server 8 having access to said files,
 - a distribution network 10, the server 8 being linked to the distribution network 10,
 - an access provider 12 for providing a client device 14 with an access to the distribution network 10.

The client device 14 has (amongst other means not represented in figure 1) a communication unit 15 for transmission/reception to/from the access provider 12, a player 16 for playing an encoded multimedia content, and a display 17 for displaying a multimedia content. The client device 14 can be either a mobile device (like a mobile phone) in which case the communication unit 15 is a radio communication unit, or a wired device (like a PC) in which case the communication unit 15 is a wired communication unit. The distribution network 10 is typically the Internet network.

For instance the broadcaster 2 is a satellite broadcasting network and the receiver 3 is a satellite receiver. This is not restrictive: any other broadcasting means could be used instead of satellite broadcasting means. The broadcasted multimedia content can be any multimedia content that is transmitted and can be received by a number of receivers including the receiver 3. For instance the broadcasted multimedia content can be a television program, a pre-recorded event/program, a live event...

The encoder 5 is responsible for encoding the received multimedia content. For instance the encoder 5 is compliant with one of the MPEG standards, or with H263.

The encoder 5 and the slicer 6 are either implemented in a single device or in two separated devices. In both cases, what is transmitted from the encoder 5 to the slicer 6 is an encoded video stream. Advantageously this encoded video stream is transmitted from the encoder 5 to the slicer 6 over IP by using the RTP protocol. This is not restrictive. By way of example, the transport layer of the MPEG-2 standard, known as MPEG-2 TS, could be used as well.

In practice, the files generated by the slicer 6 are stored in a storage unit 20 to which the server 8 has access. The storage unit 20 is shared by the slicer 6 and the server 8. The storage unit 20 can be part of the server equipment or can be located remotely.

The function of the slicer 6 is to slice the encoded content generated by the encoder 5 into a plurality of slices where each slice comprises a given amount of time of the encoded multimedia content and can be decoded independently from the other slices. In practice, any encoded multimedia content generated by a multimedia encoder comprises so-called Random Access Points (RAP). In order to produce slices that can be decoded independently one from the others, the slicer 6 slices the encoded multimedia content in such a way that each slice starts with a Random Access Point. For instance, when the encoder is compliant with the MPEG-2 or MPEG-4 standard, the random access points are the I-frames of the MPEG-encoded multimedia content, and the slicing positions are chosen in such a way that the first frame of each slice is an I-frame.

Advantageously the size of the slices is adjustable. It may be identical for all slices or it may vary from one slice to another (for instance the size of the slices may increase along time). The best efficiency is obtained with files that are relatively long because the more files to be transported the more overhead due to file headers.

Each slice generated by the slicer 6 is stored as a file in the storage unit 20. The storage unit 20 has to be "cleaned" on a regular basis to ensure that there is available room for storing the newly generated files. A way of cleaning the storage unit is to re-use file names on a regular basis. An alternative way is to use different file names for each file, and to delete the aging files on a regular basis.

According to the invention the slicer 6 generates either one set of files or a plurality of sets of files for the same multimedia content. When the slicer 6 generates a plurality of sets of files, a plurality of sets of slicing positions are used, each set of slicing positions being shifted in time compared with the other sets of slicing positions. Figure 2 is an illustration of a plurality of sets of files S_1, S_2, \dots, S_N generated by the slicer 6. Each set of files S_i comprises K files $F_{i,j}$ ($i=1, \dots, N; j=1, \dots, K$). A set of slicing positions $\{T_{i,1}, \dots, T_{i,K}\}$ is associated to each set of files S_i . As illustrated in figure 2, the slicing positions $T_{i+1,j}$ are shifted in time compared to the slicing positions $T_{i,j}$ (the axis t is the time axis). In other words, the files $F_{i+1,j}$ and $F_{i,j}$ are overlapping (they comprise identical encoded data). In figure 3, the overlap between the files $F_{i+1,j}$ and $F_{i,j}$ is indicated by an arrow O_{i+1} .

As will be apparent in the description below generating a plurality of sets of files is advantageous because it allows reducing the delay experienced by the client when he/she sends a request for a multimedia content.

The server 8 is linked to the distribution network 10. The client device 14 has access to the distribution network 10 via the access provider 12. Typically, the client device 14 can load through the distribution network 10 a page containing at least one link to one encoded multimedia content that the server 8 offers to download. When a user clicks on said link, an initial request R_0 directed to said encoded multimedia content is sent automatically to the server 8. There are several possible ways for the server 8 to handle the initial request R_0 .

In a first embodiment, the server 8 downloads a single file in response to the client request. This implementation can be used for specific applications, for instance for applications offering the client to pick-up information regarding a live event. It can also

be used with a player 16 specifically designed to cause the client device 14 to send the initial request R_0 repetitively.

In a second embodiment, the server 8 downloads the files one by one as soon as they are ready at the server side. This embodiment has the advantage of being easy to implement. But there is a risk that certain client browsers will not support receiving several files in response to one single request.

In a third embodiment (usually preferred), upon reception of the initial request R_0 , the server 8 sends a document to the client device 14. This document causes the client device 14 to repetitively send a fetching request designating the encoded multimedia content.

By way of example, the document sent by the server 8 can be a page comprising an automatic refresh command. An example of such a page is given below:

```
<html>
<head>
15 <META meta http-equiv="Refresh" content="134" ;
    url='http://www.yoursite.com/live2download.html'>
</head>
<embed src="live2download.mp4" width="240" height="240">
</embed>
20 </html>
```

Such a page causes the client browser to reload the file "live2download.mp4" every 134 seconds (which is the duration of a file in this example).

Alternatively, the document sent by the server 8 can be a standard description of the multimedia content, said standard description being intended to be processed by the player 16 in a standard way. For instance such a description can be a SMIL description (SMIL is a W3C standard defining XML-based audio/video scene descriptions). An example of such a SMIL description is given below:

```
<smil>
<head>
30 <layout>
    <root-layout width="240" height="240" background-color="white"/>
    <region regionName="im" left="0" top="0" width="240" height="240"/>
</layout>
</head>
```

```

<body>
<seq repeatCount = "indefinite" >
<video id="vid" src="live2download.mp4" region="im" />
</seq>
5 </body>
</smil>

```

The effect of this SMIL document is to cause the player 16 to play the file "live2download.mp4" repetitively. As a result the client device will repetitively send fetching requests directed to the file "live2download.mp4".

10 Advantageously, the SMIL document sent by the server 8 comprises a command indicating that the files have to be fetched some time in advance (that is some time before the end of the playback of the previous file). This ensures that the next file will arrive on time at the client device 14 so that the client will not experience a gap in the rendering of the multimedia content. An example of a SMIL description having such a

15 command is given below:

```

<smil>
<head>
<layout>
<root-layout width="240" height="240" background-color="white"/>
20 <region regionName="im" left="0" top="0" width="240" height="240"/>
</layout>
</head>
<body>
<seq repeatCount = "indefinite" >
25 <video id="vid" src="live2download1.mp4" region="im" clipBegin = "0s" dur
= "25s" />
<par>
<prefetch src="live2download2.mp4" mediaTime = "5s" />
<video id="vid" src="live2download1.mp4" region="im" clipBegin = "25s" />
30 </par>
<video id="vid" src="live2download2.mp4" region="im" clipBegin = "0s" dur
= "25s" />
<par>
<prefetch src="live2download1.mp4" mediaTime = "5s" />

```

```

<video id="vid" src="live2download2.mp4" region="im" clipBegin = "25s" />
</par>
</seq>
</body>
5 </smil>

```

This document is written for slices containing 30s of content. It causes the player to execute the following operations in sequence:

- a) playing the first 25s of a first source (live2download1.mp4);
- b) playing the last 5s of the first source and in parallel fetching the first 5s of a second
- 10 source (live2download2.mp4);
- c) playing the first 25s of the second source (which can be done without delay since the first 5s have been pre-fetched).

Using two different sources is an implementation trick. The server 8 must be designed to recognize that the first and the second sources correspond to the same encoded

15 multimedia content.

In the three embodiments described above, the server has to select a file to download upon reception of the initial request R_0 or upon reception of the fetching requests. The server 8 can either select the most recent file or the first file to get ready. The consequence of selecting the most recent file is that the client will receive out-of-

20 date data. The consequence of selecting the first file to get ready is that the client will have to wait a certain time before getting a response. In both case the inconvenience for the client is reduced when several sets of files are used. This is illustrated in figure 4.

In figure 4, three sets of files S_1 , S_2 and S_3 are represented. An arrow A indicates the reception of a request by the server 8.

25 When the only set to be generated by the slicer 6 is the first set S_1 , the server 8 will either download the file $F_{1,1}$ (the most recent file) or the file $F_{1,2}$ (the next file to get ready). If the server 8 downloads the file $F_{1,1}$, then the data received by the client will be late by a time equal to $a_{1,1}$. If the server downloads the file $F_{1,2}$, then the client will experience a delay equal to $b_{1,2}$ before receiving the data.

30 When the three sets S_1 , S_2 and S_3 are generated by the slicer 6, the server 8 will either download the file $F_{2,1}$ (the most recent file) or the file $F_{3,2}$ (the next file to get ready). If the server 8 downloads the file $F_{2,1}$, then the data received by the client will be late by a time equal to $a_{2,1}$. If the server downloads the file $F_{3,2}$, then the client will

experience a delay equal to $b_{3,2}$ before receiving the data. It can be seen that $a_{1,1} > a_{2,1}$ and $b_{1,2} > b_{3,2}$.

The transmissions over the distribution network 10 are ruled by the HTTP protocol.

5 HTTP is a stateless protocol and therefore HTTP requests issued by the same client device are normally processed independently one from the other. As a result, in the third embodiment of the invention, there is a risk that the playback of the content cannot be achieved smoothly (some parts of the content may be received several times, or some parts of the content may be missing). A fourth embodiment of the invention will now be
10 described that solves this problem.

In the fourth embodiment of the invention, the document sent by the server 8 in response to the initial request R_0 comprises a resource identifier designating the encoded multimedia content asked by the client. This resource identifier is specific to the client device 14. The document sent by the server 8 causes the client device 14 to repetitively
15 send a fetching request containing this resource identifier. Upon reception of the first fetching request, the server 8 selects the file to be downloaded as described above (it selects either the most recent file or the next file to get ready). The server 8 downloads the selected file and keeps a record of which file was downloaded (or alternatively which file is next to be downloaded). Upon reception of subsequent fetching requests
20 containing the same resource identifier, the server 8 checks the record to select the file to download, downloads the selected file and updates the record.

In this way, each client device 14 will receive a sequence of files that is complete and correctly ordered (all the received files are consecutive files belonging to the same set of files).

25 By way of example the resource identifier comprised in the document sent by the server 8 can be a "nonce" as defined in the RFC1510 of the IETF (a nonce is a number that is used only once). An example of a SMIL document comprising such a resource identifier is given below:

```

30 <smil>
    <head>
        <layout>
            <root-layout width="240" height="240" background-color="white"/>
            <region regionName="im" left="0" top="0" width="240" height="240"/>
        </layout>

```

```

</head>
<body>
<seq repeatCount = "indefinite" >
<video id="vid" src="cnn142299293873635534291919.mp4" region="im" />
5 </seq>
</body>
</smil>

```

Here the resource identifier is cnn142299293873635534291919. The client device 14 will repetitively send fetching requests for the file

10 cnn142299293873635534291919.mp4. And the server 8 will keep track of which file was downloaded (or is to be downloaded) for the resource identifier cnn142299293873635534291919.

The steps that have been discussed above are summarized in Figure 5. According to figure 5, a method according to the invention for downloading a multimedia content

15 M comprises:

- a step X1 of producing an encoded a multimedia content E(M),
- a step X2 of slicing the encoded multimedia content E(M) in at least one set of slicing positions so as to generate at least one set of slices, where each slice can be decoded independently from the other, and providing at least one set of files S₁, S₂, ..., S_N, each
- 20 file F_{i,j} (i=1,..., N; j=1,...,K) containing a slice of said encoded multimedia content, and
- a step X3 of downloading at least one of said files to the client device 14 via the distribution network 10 upon reception of an initial request R₀ directed to the encoded multimedia content E(M) from the client device 14.

These steps are implemented by way of specific hardware and/or software

25 comprised in one or several devices. For instance in figure 1, step X1 is implemented by the encoder 5, step X2 is implemented by the slicer 6, and step X3 is implemented by the server 8.

Two other examples of a network according to the invention will now be described by reference to figures 6 and 7.

30 The network of figure 6 comprises a first client device 50, a distribution network 52, a second client device 54, and at least one access provider 56 for providing the first and the second client devices 50 and 54 with an access to the distribution network 52. The second client device 54 is similar to the client device 14 described by reference to figure 1. Typically the distribution network 52 is the Internet network.

The first client device 50 comprises:

- a source 60 for acquiring a multimedia content,
- an encoder 62 for encoding an acquired multimedia content,
- a slicer 64 for slicing an encoded multimedia content in at least one set of slicing
- 5 positions so as generate at least one set of slices where each slice can be decoded independently from the others, and for providing at least one set of files, each file containing a slice of said encoded multimedia content,
- a server 66 having access to said files,
- a communication unit 68 for transmission/reception to/from the access provider 56.

10 Typically the first client device is a mobile phone, which means that the communication unit 68 is a radio communication unit.

The functionalities of the source 60, the encoder 62, the slicer 64 and the server 66 are identical to the functionalities of the source 1, the encoder 5, the slicer 6 and the server 8 described above by reference to figures 1 to 4.

15 Figure 7 gives a schematic representation of an alternative solution where the server 66 is located in the distribution network 52 instead of being located in the first client device 50. In this embodiment, the first client device 50 will upload the files generated by the slicer 64 to the server 66, and the server 66 will in turn download at least one the files to the second client device 54.

20 Typically the first client 50 sends a link toward an encoded multimedia content (for instance a video that has just been captured by the first client device 50) to the second client device 54, for instance via SMS (Short Message Service). When the second client clicks on the link contained in the SMS, an initial request directed to the encoded multimedia content is sent to the first client device 50. Upon reception of this

25 initial request, the first client device 50 operates as described above with reference to figures 1 to 4.

With respect to the described network, server, system, slicer, client device, and downloading method, modifications or improvements may be proposed without departing from the scope of the invention. The invention is thus not limited to the

30 examples provided.

In particular the "progressive downloading" concept disclosed in the European patent application n° 03290453.4 filed on May 07, 2003 by Koninklijke Philips Electronics N.V. can be combined with present invention. When a file generated by the

slicer 6 is progressively downloaded, the player 16 doesn't need to wait until the file is completely downloaded to start playing the file.

The use of the verb "comprise" in the description and in the claims does not exclude the presence of other elements than those listed in the description and in the claims.

CLAIMS

1. A network comprising at least:
 - a source (1) for acquiring a multimedia content,
 - an encoder (5) encoding said multimedia content,
 - a slicer (6) for slicing said encoded multimedia content in at least one set of slicing
 - 5 positions so as generate at least one set of slices where each slice can be decoded independently from the others, and for providing at least one set of files, each file containing a slice of said encoded multimedia content,
 - a distribution network (10),
 - an access provider (12) for providing a client device (14) with an access to said
 - 10 distribution network,
 - a server (8) linked to said distribution network and having access to said set(s) of files for downloading at least one of said files to said client device via said distribution network upon reception of an initial request from said client device, said request being directed to said encoded multimedia content.
 - 15
2. A system (4) comprising:
 - an encoder (5) for encoding a multimedia content, thereby generating an encoded multimedia content,
 - and a slicer (6) for slicing said encoded multimedia content in at least one set of slicing
 - 20 positions thereby generating at least one set of slices where each slice can be decoded independently from the other, and for providing at least one set of files, each file containing a slice of said encoded multimedia content.
3. A slicer (6) for slicing an encoded multimedia content in at least one set of slicing
- 25 positions thereby generating at least one set of slices where each slice can be decoded independently from the other, and for providing at least one set of files, each file containing a slice of said encoded multimedia content.
4. A server (8) having access to at least one set of files (S_i) generated by slicing an
- 30 encoded multimedia content in at least one set of slicing positions ($\{T_{i,1}, \dots, T_{i,K}\}$) so as to form slices that can be decoded independently one from the other, each file ($F_{i,j}$) containing a slice of said encoded multimedia content, said server having means for

downloading to a client device (14) at least one of said files upon reception of a request directed to said encoded multimedia content from said client device.

5. A server as claimed in claim 4, wherein said downloading means comprise:

- 5 - means for sending a document to said client device upon reception of said request, said document causing said client device to repetitively send a fetching request designating said encoded multimedia content,
- means for selecting which file is to be downloaded amongst said set(s) of files, upon reception of said fetching requests from said client device,
- 10 - means for downloading the selected file.

6. A server as claimed in claim 4, wherein said downloading means comprise:

- means for sending a document to said client device upon reception of said initial request, said document comprising a resource identifier designating said encoded
- 15 multimedia content and specific to said client device, said document causing said client device to repetitively send a fetching request containing said resource identifier,
- means, activated upon reception of a first fetching request, for:
selecting a first file amongst the set(s) of files generated from said encoded multimedia content, downloading the selected file, and keeping a record of said resource identifier
- 20 together with an indication of which file was downloaded,
- means, activated upon reception of other subsequent fetching requests, for:
checking said record so as to select the file that follows the one that was previously downloaded, downloading the selected file, and updating said record.

25 7. A sever as claimed in one of claims 5 or 6, wherein said document comprises an instruction for the client device to send a subsequent fetching request before the end of the playback of the file that was downloaded in response to the previous fetching request.

30 8. A client device having:

- means for connecting to a server, said server having access to at least one set of files generated by slicing an encoded multimedia content in at least one set of slicing positions so as to form slices that can be decoded independently one from the other, each file containing a slice of said encoded multimedia content, said server offering to

download at least one of said files,

- means for sending to said server a request directed to said encoded multimedia content,

- means for receiving one of said files,

- means for playing the received file,

5 - means for sending at least one subsequent request directed to the same encoded multimedia content to said server.

9. A client device as claimed in claim 8, further comprising means for sending said subsequent request before the end of the playback of the received file.

10

10. A method for downloading an encoded multimedia content to a client device, said method comprising the steps of:

- encoding a multimedia content,

15 - slicing said encoded multimedia content in at least one set of slicing positions so as generate at least one set of slices, where each slice can be decoded independently from the other, and providing at least one set of files, each file containing a slice of said encoded multimedia content, and

20 - downloading at least one of said files to said client device via a distribution network upon reception of an initial request directed to said encoded multimedia content from said client device.

ABSTRACT**METHOD FOR DOWNLOADING A BROADCASTED MULTIMEDIA CONTENT
OVER A DISTRIBUTION NETWORK**

The invention proposes to transform a broadcasted multimedia content (for instance a broadcasted TV program or a live event) into a file-based content. More specifically, it consists in slicing an encoded content into a plurality of slices where each slice comprises a given amount of time of the content and can be decoded independently
5 from the other slices. Each slice is stored in a file that can be downloaded as soon as it is ready at the server side and can be played as soon as it is received at the client side.

The transmission between the server and the client device is ruled by the HTTP protocol that is accepted by all firewalls and NAT. The consequence is that the transmitted content is accessible for any client device that has access to the Web without
10 restriction

Application: downloading a broadcasted content via the internet.

Reference: figure 1

1/3

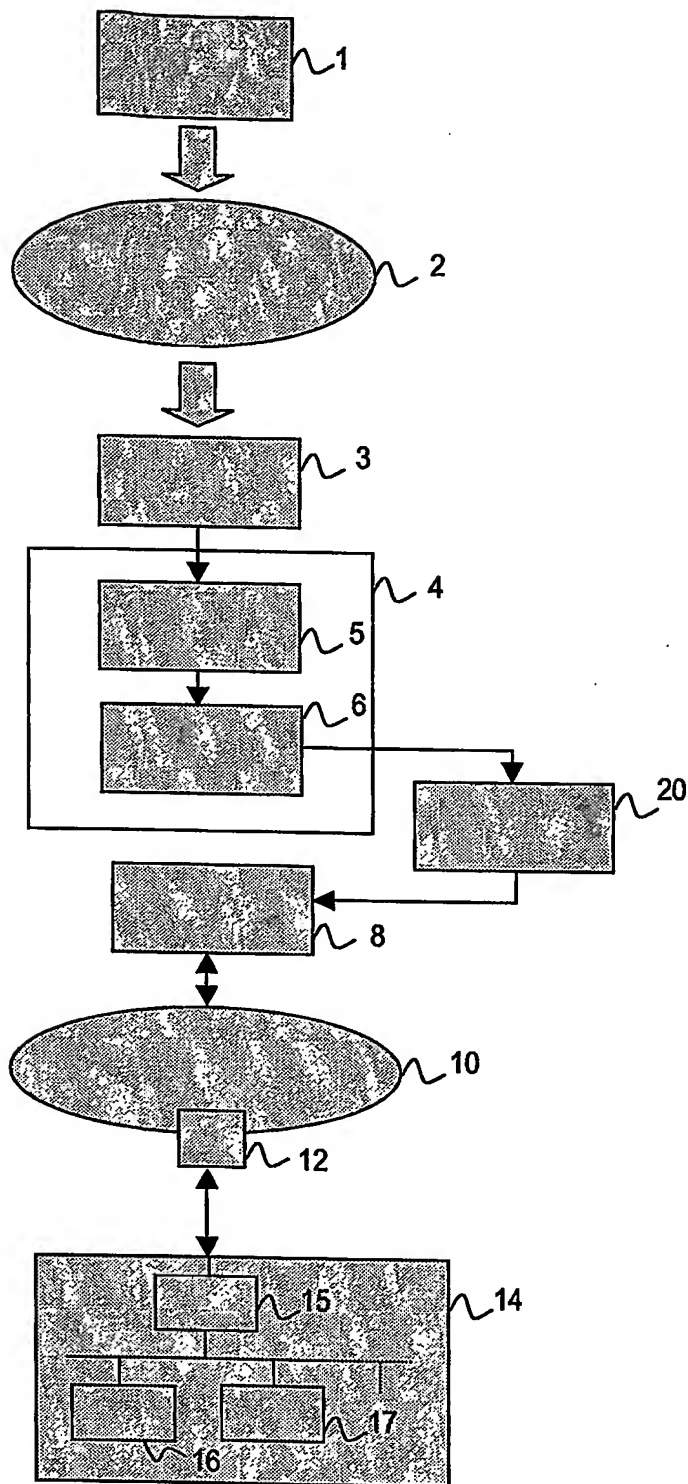


FIG. 1

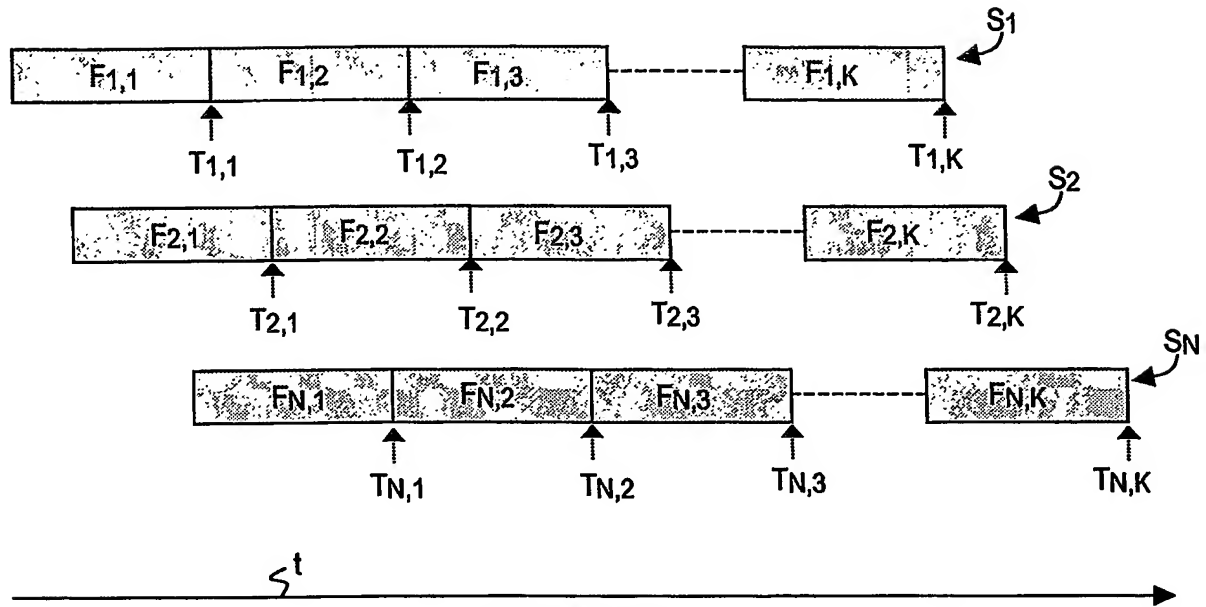


FIG.2

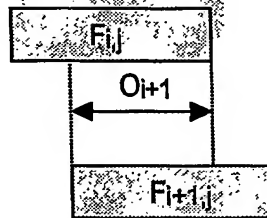


FIG.3

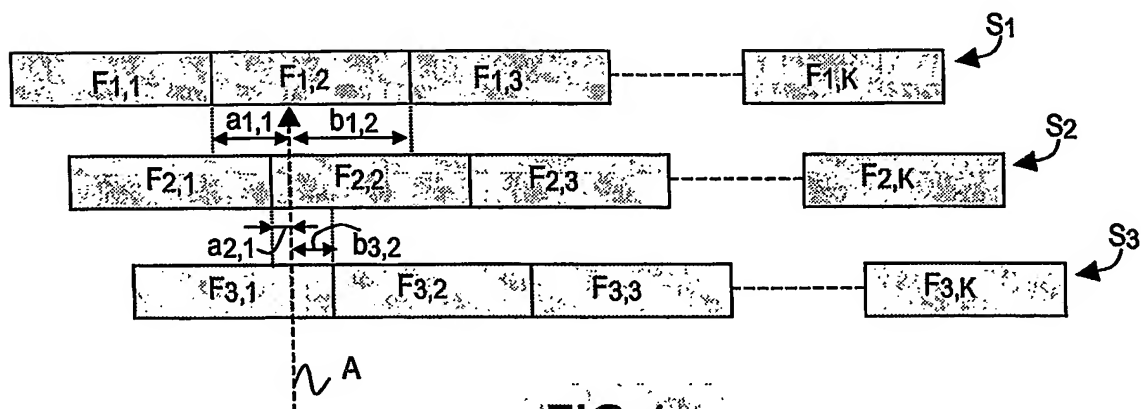


FIG.4

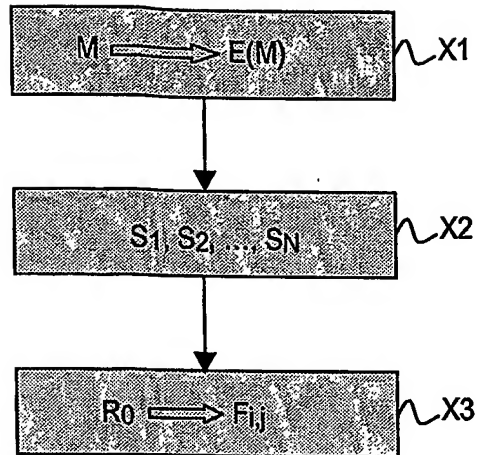


FIG.5

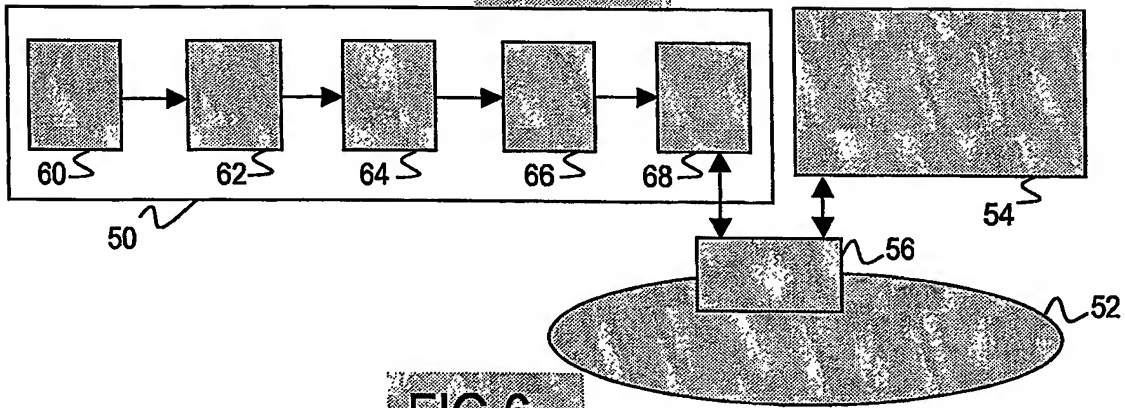


FIG.6

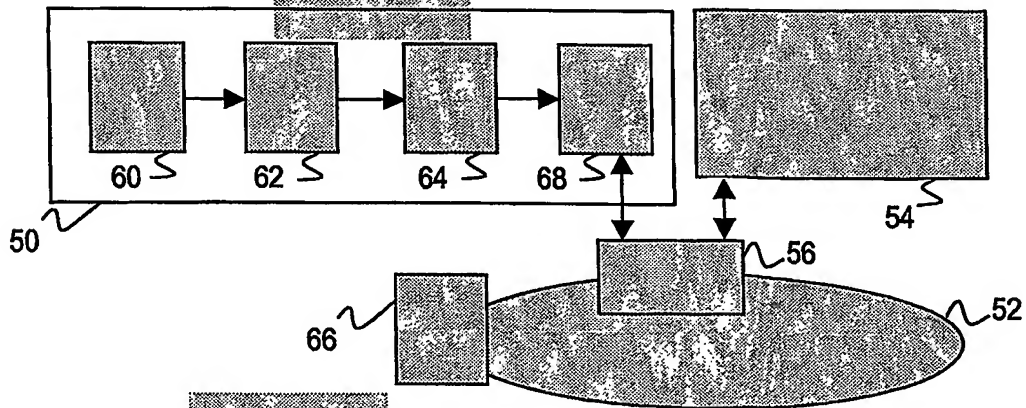


FIG.7

FCIT/IB004/002141



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER: _____**

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.